

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平10-154116

(43)公開日 平成10年(1998) 6 月 9 日

(51)Int.Cl.⁹

G 0 6 F 13/00

識別記号

3 5 3

3 5 1

F I

G 0 6 F 13/00

3 5 3 S

3 5 1 E

審査請求 未請求 請求項の数 9 O L (全 18 頁)

(21)出願番号

特願平8-313805

(22)出願日

平成 8 年(1996)11月25日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目 6 番地

(72)発明者 山▲崎▼ 康雄

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 森 利明

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 鵜飼 敏之

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(74)代理人 弁理士 高橋 明夫

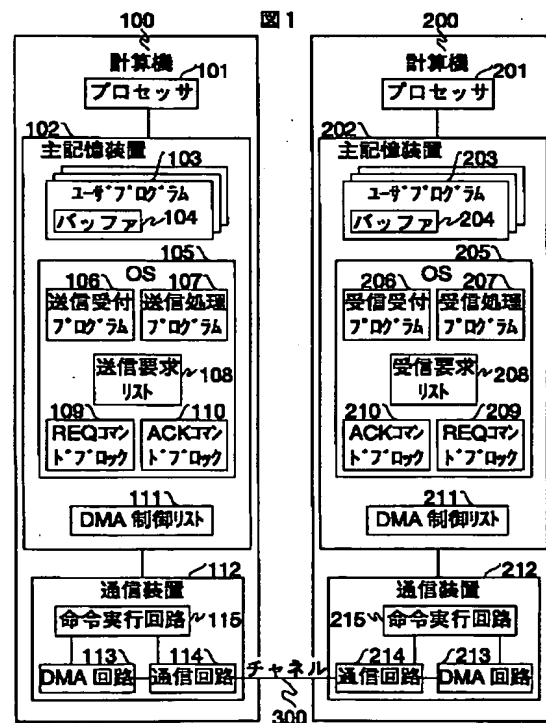
最終頁に続く

(54)【発明の名称】 データ転送方法

(57)【要約】

【課題】 OS により送信データをバッファリングしないで転送する。

【解決手段】 送信側の計算機 100 では、OS 105 は、複数のユーザプログラム 103 から指定された複数の送信データの送信先ユーザプログラム 203 がそれぞれデータを受信可能な状態にあるかを OS 205 と通信してまとめて確認する。受信側のユーザプログラム 204 が受信要求を発行していない場合には、そのプログラムは受信不可とし、送信可と判断された複数の送信データの送信をまとめて通信装置 112 に指示する。通信装置 112 では、DMA 回路 113 が各ユーザプログラム 103 のバッファ 104 から送信データを直接連続して読み出し、通信回路 114 がそれぞれの送信データをチャネル 300 を介して送信する。受信側の計算機 200 では、DMA 回路 213 が各送信データを送信先のユーザプログラム 104 のバッファ 204 に直接連続して書き込む。



【特許請求の範囲】

【請求項1】 (a) 第1の計算機の送信元のユーザプログラムにより、該第1の計算機の主記憶装置上のその送信元のユーザプログラム用の第1の領域に保持されたデータを、その第1の計算機に伝送路を介して接続された第2の計算機の送信先のユーザプログラムへ送信することを要求する送信要求をその第1の計算機を制御する第1のOSに発行し、(b) 該送信要求に応答して、該第1のOSから該第2の計算機を制御する第2のOSに、該送信先のユーザプログラムがデータ受信可能な状態にあるか否かを問い合わせ、(c) 該問い合わせに対して、該第2のOSが該送信先のユーザプログラムがデータ受信可能な状態にあるか否かを判断し、その判別結果を示す問い合わせ結果を該第2のOSにより該第1のOSに通知し、(d) 該第2のOSが該送信先のユーザプログラムがデータ受信可能な状態にあると判断したときに、該第2のOSにより、該送信要求が指定したデータの受信と、該第2の計算機の主記憶装置上に定められた該送信先のユーザプログラム用の第2の領域へのそのデータの書き込みとを該第2の計算機の通信装置に指示し、(e) 該問い合わせ結果が該送信先のユーザプログラムがデータ受信可能な状態にあることを示すとき、該第1のOSにより上記データの送信を該第1の計算機の通信装置に指示し、(f) 該第1のOSによる該指示に応答して、該第1の計算機の該通信装置により該データを該第1の領域から読み出し、該第2の計算機に該伝送路を介して転送し、(g) 該第2のOSによる該指示に応答して、該第2の計算機の該通信装置により、該データを受信し、該第2の領域に書き込むステップからなるデータ転送方法。

【請求項2】 該問い合わせ結果が、該送信先のユーザプログラムがデータ受信不可能な状態にあることを示すとき、該第1のOSにより、該ステップ(b)を繰り返すステップをさらに有する請求項1記載のデータ転送方法。

【請求項3】 該送信先のユーザプログラムから該データの受信要求を該第2のOSに要求するステップをさらに有し、
該ステップ(c)は、
該問い合わせを受けた時点で、該送信先のユーザプログラムから該データの受信要求がすでに発行されているか否かに応じて、該送信先のユーザプログラムが受信可能な状態にあるか否かを判別するステップを有する請求項1または2記載のデータ転送方法。

【請求項4】 上記送信要求は、上記データのサイズを指定し、
上記受信要求は、該送信先のユーザプログラム用の上記領域のサイズを指定し、
ステップ(c)は、
該問い合わせを受けた時点で、該送信先のユーザプロ

ラムから該データの受信要求がすでに発行されているときに、上記データのサイズが該第2の領域のサイズを越えないか否かをさらに判別し、
該問い合わせを受けた時点で、該送信先のユーザプログラムから該データの受信要求がすでに発行されているときでも、上記データの上記サイズが該第2の領域のサイズを超えているときには該送信先のユーザプログラムが受信不可能な状態にあると判別するステップを有する請求項3記載のデータ転送方法。

- 10 【請求項5】 上記ステップ(a)は、該複数の送信元のユーザプログラムにより該第2の計算機で実行中の複数の送信先のユーザプログラムに対して複数の送信要求が発行されるように、該第1の計算機で実行中の複数の送信元のユーザプログラムの各々により実行され、
上記ステップ(b)は、該複数の送信要求が指定する該複数の送信先のユーザプログラムがそれぞれデータ受信可能な状態にあるか否かを該第1のOSから該第2のOSにまとめて問い合わせるステップからなり、
上記ステップ(c)は、該第2のOSが、該複数の送信先のユーザプログラムがそれぞれデータを受信可能な状態にあるか否かを判断し、それぞれの送信先のユーザプログラムに対する判別結果を示す複数の問い合わせ結果を該第2のOSにより該第1のOSにまとめて通知するステップからなり、
上記ステップ(d)は、該第2のOSにより、該複数の送信先のユーザプログラムの内の一部の複数の送信先のユーザプログラムがデータを受信可能な状態にあると判断されたときに、該一部の複数の送信先のユーザプログラムに対する複数の送信要求が指定した一部の複数のデータを、該第2の計算機の該主記憶装置上に定められた該一部の該複数の送信先のユーザプログラム用の一部の複数の領域へそれぞれ書き込むことを該第2の計算機の該通信装置にまとめて指示するステップからなり、
上記ステップ(e)は、該複数の問い合わせ結果に応答して、該第1のOSにより、該第1の計算機の該主記憶装置上に保持された該一部の複数の送信元のユーザプログラムが指定した一部の複数のデータの送信を該第1の計算機の該通信装置にまとめて指示するステップからなり、
40 上記ステップ(f)は、該第1のOSによる該指示に応答して、該第1の計算機の該通信装置により、該一部の複数のデータを該第1の計算機の該主記憶装置から順次読み出し、該第2の計算機に該伝送路を介して順次転送するステップからなり、
上記ステップ(g)は、該第2の計算機の該通信装置により、該一部の複数のデータを順次受信し、かつ、該第2の計算機の該主記憶装置上の該一部の複数の領域に書き込むステップからなる請求項1記載のデータ転送方法。
- 50 【請求項6】 該複数の問い合わせ結果のいずれかが少な

くとも一つの他の送信元ユーザプログラムがデータ受信不可能な状態にあることを示すとき、該第1のOSにより、該少なくとも一つの他の送信元ユーザプログラムが発行した送信要求に関して該ステップ(b)を繰り返すステップをさらに有する請求項5記載のデータ転送方法。

【請求項7】上記ステップ(b)は、一定時間間隔で繰り返し実行される請求項5記載のデータ転送方法。

【請求項8】該複数の送信先のユーザプログラムのそれぞれからデータの受信要求を該第2のOSに要求するステップをさらに有し、

該ステップ(c)は、

該問い合わせを受けた時点で、該複数の送信先のユーザプログラムの各々からデータの受信要求がすでに発行されているか否かに応じて、その送信先のユーザプログラムが受信可能な状態にあるか否かを判別するステップを有する請求項5または6記載のデータ転送方法。

【請求項9】上記複数の送信要求は、それぞれ送信すべきデータのサイズを指定し、

上記各受信要求は、該第2の計算機の該主記憶装置上に形成されたその受信要求を発行した送信先のユーザプログラム用の領域のサイズを指定し、

ステップ(c)は、

該問い合わせを受けた時点で、各送信要求が要求する送信先のユーザプログラムからデータの受信要求がすでに発行されているときに、その送信要求が指定するデータのサイズがその受信要求が指定したその送信先プログラム用の上記領域のサイズを超えないか否かをさらに判別し、

該問い合わせを受けた時点で、各送信要求が指定する送信先のユーザプログラムからデータの受信要求がすでに発行されているときでも、その送信要求が指定する上記データの上記サイズがその送信先のユーザプログラム用の上記領域のサイズを超えているときにはその送信先のユーザプログラムが受信不可能な状態にあると判別するステップを有する請求項8記載のデータ転送方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は伝送路で接続された計算機間のデータ通信方法に係る。

【0002】

【従来の技術】従来の計算機システム間でのデータ転送方法としては、ハイレベル・データリンク制御手順(HDLC: High Level Data Link Control)がある。例えば、“データ通信”、データ通信教育委員会編、共立出版、pp. 46-53(1982)に示す「ハイレベル・データリンク制御手順(HDLC: High Level Data Link Control)」参照。HDLCは送信側の計算機が受信側の計算機に転送の通知を出し、受信側の計

算機から応答が返ってきてからデータ転送を行う、というデータ転送方法である。このデータ転送方法は、直接データ転送路で接続された計算機間のデータ転送に適用できる。このHDLC手順は、ユーザプログラム間でデータを直接転送することができるので高速転送に向いているが、それぞれOSで制御される任意のユーザプログラム間のデータ転送には適用できない。なお、HDLC手順によるデータ転送を高速化するために、送信側計算機の送信回路に、送信側のプログラムのバッファ内の送信データを連続して読み出すDMA回路を設け、受信側計算機の受信回路に受信したデータを受信側のプログラムのバッファに連続して書き込むDMA回路を設ける工夫もなされている。

【0003】それぞれオペレーティングシステム(OS)で制御される任意のユーザプログラム間のデータ転送の方法としては、TCP/IP(Transmission Control Protocol/Internet Protocol)による方法がある。例えば、中村明/相田仁/計字生/小池汎平共訳「UNIX 4.3BSDの設計と実装」丸善、pp. 169-185(1991)(原著: The Design and Implementation of the 4.3BSD Operating System、原著者: S. J. Leffler/M. K. McKusick/M. J. Karels/J. S. Quarterman、原発行所Addison Wesley)参照。このデータ転送方法は、計算機ネットワークで接続された任意の計算機間での、他の計算機を介したパケットによるデータ転送にも使用できる。

【0004】この方法では、複数のユーザプログラムが同時にデータを転送できるようにOSがバッファリングを行っている。すなわち、送信側の計算機では、OSは、ユーザプログラムに割り当てられたバッファ領域からそのOS内バッファへ送信すべきデータをコピーする。この際、コピーしたデータにパケットヘッダを付加する。この送信データとヘッダを含むパケットを送信回路によりネットワークを介して受信側計算機に送信する。受信側計算機では、受信回路が、受信したパケットを受信側OS内のバッファに格納し、受信側OSは、書き込まれたパケットのヘッダから受信先のユーザプログラムを判別し、その受信側ユーザプログラムから受信要求が来たときに、その受信側のプログラムのバッファへ、受信側OSのバッファ内にある受信したパケット内のデータをコピーする。この時点で受信側のOSは受信完了を送信側のOSに通知する。このデータ転送方法では、一つのパケットで送信されるデータの長さは、送信側OSおよび受信側OS用のバッファのサイズに固定されている。このバッファの大きさを超えるデータを転送する場合にはこのバッファのサイズでもってそのデータを分割し、分割された複数の部分データを複数のパケッ

トにより転送する。送信側計算機のOSおよび受信側計算機のOSでは、通信装置の起動処理、終了処理をそれぞれのバケットに対して行う。

【0005】この転送方法では、受信側のOS内のバッファに、受信データを一時的に保持するので、受信側のユーザプログラムは、上記データの送信と非同期に受信要求を発行できる。また、もし、上記データの送信が不成功となったときには、送信側のOSがそのOS用のバッファ内の送信データを再送する。このため、送信側のユーザプログラムは、受信側のユーザプログラムが受信可能な状態になっているか否かによらないでデータの送信を送信側のOSに要求でき、しかも、そのデータがそのOS内のバッファにコピーされた後は、そのデータの転送が完了するのを待たないで他の処理を実行できる。

【0006】この方法によるデータ転送を高速に行うために、送信側計算機の送信装置内に、送信側のOSのバッファから送信データを連続して読み出すためのDMA回路を設け、受信側計算機の受信装置内に、受信したデータを受信側OSのバッファに連続して書き込むためのDMA回路を設けるという工夫も行われている。例えば、山居正幸著「イーサネット・ボード&カード」、オープンデザイン、No. 3、CQ出版、pp. 109-113 (1994) 参照。

【0007】

【発明が解決しようとする課題】TCP/IPによるデータ転送では、送信側の計算機および受信側の計算機のいずれにおいてもユーザプログラムのバッファとOSのバッファの間でデータコピーが行われる。このデータコピー処理にかかる時間はデータ量に比例して大きくなり、このデータコピー時間が、データ転送時間に占める割合が大となる。その結果、計算機間を高速な伝送路で接続してもこの方法によるデータ転送速度はさほど改善されない問題点がある。前述のように、送信側計算機の送信回路と受信側計算機の受信回路の両方にDMA回路を設けてもこの問題は残る。とくに科学計算分野では、非常に多量のデータを計算機間で連続して転送する必要が生じ、このデータコピー時間の問題が著しく大となる。

【0008】さらに、この方法では、送信側OSおよび受信側OS用のバッファのサイズを超えるデータを転送する場合、前述のように、そのデータを分割して得られる複数の部分データを複数のバケットにより転送する。この際、送信側計算機のOSおよび受信側計算機のOSでは、通信装置の起動処理、終了処理をそれぞれのバケットに対して行う。このデータのサイズが大きくなると、それを転送するのに必要なバケット数が多くなり、これらのOSが行う通信装置の起動処理、終了処理に時間が掛かり、それによりデータ転送速度が低減するという問題がある。この問題も科学計算分野のごとく非常に多量のデータを計算機間で連続して転送する場合に大き

くなる。

【0009】TCP/IPによるデータ転送では、多数の計算機を経由したデータ転送にも使用できるが、その経由のためにデータ転送速度は低下する。科学技術計算用のシステムでは、伝送路で結合された特定の計算機間で他の計算機を介さないでより高速にデータを転送することが望まれる。

【0010】従って、本発明の目的は、送信側計算機のOSで制御されるユーザプログラムから、受信側計算機内のOSで制御されるユーザプログラムに、より高速にデータを転送するデータ転送方法、とくに、科学計算用のユーザプログラムの間で非常に多量のデータを他の計算機を介しないで転送するのに適したデータ転送方法を提供することである。

【0011】本発明のより具体的な目的は、送信側計算機でのユーザプログラム用のバッファとOS用のバッファ間のデータコピーおよび受信側計算機でのユーザプログラム用のバッファとOS用のバッファ間のデータコピーを行わないで、それらのユーザプログラム間でデータを転送するデータ転送方法を提供することにある。

【0012】本発明の他の具体的な目的は、送信側計算機のユーザプログラムが指定したデータを分割しないで受信計算機のユーザプログラムに転送するデータ転送方法を提供することにある。

【0013】

【課題を解決するための手段】上記目的を達成するために、本発明によるデータ転送は、相互に伝送路で接続された第1、第2の計算機を有し、該第1、第2の計算機がプロセッサと主記憶装置と該主記憶装置上に格納されたデータを読み出し、該伝送路に送信しあるいは該伝送路からデータを受信し、該主記憶装置に書き込む通信装置を有するシステムにおいて実行される。まず、第1の計算機で実行されている送信元のユーザプログラムと第2の計算機で実行されている送信先のユーザプログラムの間でデータ転送が可能か否かを第1の計算機を制御する第1のOSと第2の計算機を制御する第2のOSとの間の通信により判別する。もしそれらのプログラム間でデータ転送が可能と判断されたときには、それらのOSからの指示に従って、送信側計算機の通信装置と受信側計算機の通信装置によりそれらのユーザプログラムの間でデータをそれらのOSを介しないで直接転送する。より具体的には、(a) 送信元のユーザプログラムにより、その送信元のユーザプログラム用の第1の領域に保持されたデータを第2の計算機で実行されている送信先のユーザプログラムへ送信することを要求する送信要求を第1のOSに発行し、(b) 第1のOSから第2のOSに、送信先のユーザプログラムがデータ受信可能な状態にあるか否かを問い合わせ、(c) 第2のOSが送信先のユーザプログラムがデータ受信可能な状態にあるか否かを判断し、その判別結果を示す問い合わせ結果を第

2のOSにより第1のOSに通知し、(d)第2のOSが送信先のユーザプログラムがデータ受信可能な状態にあると判断したときに、第2のOSにより、送信要求が指定したデータの受信と、送信先のユーザプログラム用の第2の領域へのそのデータの書き込みとを第2の計算機の通信装置に指示し、(e)問い合わせ結果が送信先のユーザプログラムがデータ受信可能な状態にあることを示すとき、送信側計算機のOSにより上記データの送信を送信側計算機の通信装置に指示し、(f)第1の計算機の通信装置によりデータを送信元ユーザプログラム用の第1の領域から読み出し、第2の計算機に伝送路を介して転送し、(g)第2の計算機の該通信装置により、データを受信し、送信先のユーザプログラム用の上記第2の領域に書き込む。

【0014】

【発明の実施の形態】以下、本発明に係るデータ転送方法を図面に示した実施の形態を参照してさらに詳細に説明する。

【0015】<発明の実施の形態>

(1) システム構成と動作の概要

図1に示した計算機システムにおいて、計算機100、200はそれぞれプロセッサ101、201と、主記憶装置102、202と、通信装置112、212とからなり、これらの通信装置が高速かつ高信頼な伝送路であるチャンネル300で接続され、これらの計算機は、このチャンネル300を介して他の計算機を介することなく直接に接続されている。通信装置112、212は、それぞれダイレクトメモリアクセス(DMA)回路113、213および通信回路114、214、命令実行回路115、215からなり、DMA回路113、213はそれぞれ主記憶装置102、202上の領域を直接読み書き可能になっている。OS105、205が通信装置112、212への命令を主記憶装置102、202上に用意されたDMA制御リスト111、211に格納して通信装置112、212を起動すると、命令実行回路115、215がDMA制御リスト111、211に格納された命令を解釈し、DMA回路113、213および通信回路114、214を制御することで与えられた命令を実行し、複数の命令がある場合は残りの命令も同様に順次実行し、全ての命令の解釈し、その実行が完了すると、OS105、205に対してハードウェア割り込みを発生して命令の完了報告を行う。DMA制御リスト111、211に格納する命令は、(1)主記憶装置112、212のどこから、(2)どれだけの大きさを、(3)送信するかあるいは受信するか、という指定の組からなる任意の個数のDMA制御命令のリストである。通信装置112、212の双方で対となるDMA制御命令の組が実行されると通信路300を介した通信が成立する。対となるDMA制御命令の組とは、同一の大きさの送信DMA制御命令と受信DMA制御命令の組であ

る。例えば計算機100において命令実行回路115が「主記憶装置112のアドレスAより、大きさaだけ送信」という命令を解釈すると、DMA回路113に対しては指定された領域(アドレスAより大きさa)を読み出してその結果を通信回路112に渡すように指示し、通信回路112に対しては渡されたデータを伝送路300に出力するように指示する。その場合、計算機200において命令実行回路215に与えられる命令は「主記憶装置212のアドレスBより、大きさaだけ受信」でなければならない。そして、通信回路112の送信動作と通信回路212の受信動作の双方が行われた時点より伝送路300を介した通信が開始される。通信回路112、212は、送信動作あるいは受信動作のどちらかいずれかが先に発生した場合は、受信動作あるいは送信動作が発生するのを待つ。

【0016】主記憶装置102、202において、計算機100、200を制御するOS105、205の制御下で複数のユーザプログラム103、203が動作している。各々のユーザプログラム103、203には各計算機内で一意なプログラム識別子が割り当てられており、OS105、205はこのプログラム識別子で指定して、ユーザプログラムの動作を中断し、あるいは再開することができる。

【0017】複数のユーザプログラム103の各々は、送信データをバッファ104に格納した後に、OS105に送信要求を発行する。OS105は、各ユーザプログラム103内のバッファ104に格納された送信データを受信側計算機200に転送することを通信装置112に指示する。通信装置112では、従来技術と異なり、DMA回路113が各ユーザプログラム103内のバッファ104から送信データを直接連続して読み出す。OS105は、各ユーザプログラム103が指示した送信データの送信を通信装置112に指示するが、従来技術と異なり、その送信データをそのOS105内のバッファにコピーしない。通信回路114が、DMA回路113が読み出した送信データをチャンネル300を介して受信側の計算機200に送信する。

【0018】受信側の計算機200の通信装置212では、通信装置214は、それぞれの送信データを受信するごとに、従来技術と異なり、DMA回路213が各送信データの送信先のユーザプログラム104内のバッファ204に受信されたデータを、受信側計算機を制御するOS205を介することなく直接連続して書き込む。OS205は、従来技術と異なり、受信データをそのOS205内のバッファにコピーしない。

【0019】上記の方法によりデータを転送するには、受信側のユーザプログラム203と上記データ転送との同期を取る必要がある。すなわち、複数のユーザプログラム103のそれぞれから指定された送信データの送信先のユーザプログラム203がデータを受信可能な状態

にしなければならない。ユーザプログラム203が受信可能な状態とは受信したデータを蓄えるためのバッファ204が用意されているということである。このバッファ204を繰り返し使用する場合などにバッファ204の上書きによるデータ破壊を防止するため、受信ユーザプログラム203がOS205に対して明示的に受信要求を発行することによって、受信準備が整ったことすなわち受信可能な状態であることを通知する。受信要求の発行は後述する受信受付プログラム206の呼び出しで行う。このために、本実施の形態では、OS105は、複数のユーザプログラム103から指定された送信データの送信先のユーザプログラム203がデータを受信可能な状態にあるかをOS205と通信してまとめて確認し、この確認後にそれらのユーザプログラム103から指定された複数の送信データの送信をまとめて通信装置112に指示する。

【0020】具体的には、OS105内の送信受付プログラム106が、複数のユーザプログラム103から発行された複数の送信要求を送信要求リスト108に蓄え、送信処理プログラム107が一定の時間間隔ごとに、それらの蓄えられた送信要求を表すREQコマンドブロック109を作成し、さらに、DMA制御リスト111を生成して通信装置112にREQコマンドブロック109の送信を指示する。通信装置112が、このブロック109を計算機200に送信すると、計算機200では、いずれかのユーザプログラム203が計算機100内のいずれかのユーザプログラム103からデータを受信すべき状態になると、データ受信要求をOS205に発行するように構成され、OS205内の受信受付プログラム206は、この受信要求を受信要求リスト208に格納する。

【0021】通信回路214がこのREQコマンドブロック109を受信すると、DMA回路213が主記憶装置202にREQコマンドブロック209として書き込み、受信処理プログラム207は、受信要求リスト208に保持された受信要求と書き込まれたREQコマンドブロック209とに基づいて、このREQコマンドブロック209が指定する送信先のユーザプログラムが受信可能な状態にあるか否かをそれぞれ示すACKコマンドあるいはNACKコマンドを含むACKコマンドブロック210を作成し、さらに、DMA制御リスト211を生成して通信装置212にそのACKコマンドブロック210の送信を指示する。計算機100では、ACKコマンドブロック210を受信すると、送信処理プログラム107が、このACKコマンドブロック210に基づいて、先に送信したREQコマンドブロック109内の送信要求の内、送信可能なもののみに付随するデータの送信を通信装置112に指示する。

【0022】なお、OS105、205は同じ構造を有し、いずれの計算機も他方の計算機に対して送信および

受信を行える。従って、計算機100内のOS105には、受信受付プログラム206、受信処理プログラム207、受信要求リスト208を有し、計算機200内のOS205には、送信受付プログラム106、送信処理プログラム107、送信要求リスト108を有する。しかし、図では簡単化のために、計算機100が送信側の計算機、計算機200が受信側の計算機として動作する場合に必要な部分のみを図示してある。以下、本実施の形態におけるデータ転送方法の詳細を、図2に示す主要部分の動作のタイムチャートを参照して説明する。なお、以下では、バッファ104と204、REQコマンドブロック109、209、ACKコマンドブロック210、110とDMA制御リスト111、211は主記憶装置102あるいは202上で連続して常駐しているものと仮定する。領域が「連続する」とは仮想的な記憶上ではなく物理的な主記憶上において連続して領域が確保されているという意味であり、領域が「常駐する」とは通常のユーザプログラムの扱う記憶領域とは異なり確保された領域が必要に応じてOS105、205によりディスク装置に待避されることがなく常に物理的な主記憶上に存在し続けるということを意味する。これは通信装置112、212からは主記憶装置102、202が仮想的な領域として認識できないためである。また、全てのコマンドブロック109、210、110、209は等しい大きさであると仮定する。転送するファイルの数やそれぞれのファイルの大きさなどについての情報は、送信側と受信側のユーザプログラム間で予め明らかであるとする。

【0023】(2) データ転送動作の詳細

(A) 送信側計算機100 (その1) REQコマンドの送信

(A1) 送信側のユーザプログラム103からの送信要求の発行

送信側の計算機100において、三つの送信側ユーザプログラム103が走行し、受信側の計算機でも三つの受信側ユーザプログラム203が走行していると仮定する。

【0024】図3のフローチャートを参照するに、三つの送信側ユーザプログラム103の各々の処理は、2重のループ構造となっており、処理の最初と最後に転送すべきファイルを保持するための送信用のバッファ104を確保し、あるいは解放する(ステップ401、408)。各ユーザプログラム用のバッファ104は、それが確保された後解放されるまでの間は主記憶装置102上に常駐し、全処理中に渡って繰り返し使用されるものとする。外側のループは複数のファイルを処理するためのループであり、内側のループは各々のファイルを複数回の通信で転送するためのループである。

【0025】各送信側ユーザプログラム103は、バッファ104に転送するファイルの一部を格納すること

10

20

30

40

50

(ステップ402)、送信受付プログラム106を呼び出し、送信要求を送出する(ステップ403)。送信受付プログラム106を呼び出した結果がエラーであればデータの送信は失敗であり、ハードウェアのエラーまたはソフトウェアのエラーであるので、エラー処理を行い終了する(ステップ405)。バッファ104内のデータ送信を成功裡に終わると、現在処理中のファイルが全て転送済みであるか調べて未処理の領域があればそれら进行处理するためにステップ402から繰り返す(ステップ406)。各ユーザプログラムは送信すべきファイルがまだある限り以上のステップを繰り返す(ステップ407)。このように各送信ユーザプログラム103は固定長のバッファ104を使いまわすようになっている。同様に各受信ユーザプログラム203も固定長のバッファ204を使いまわすようになっている。本発明を用いたデータ転送においては任意のサイズのデータを転送することが可能であるが、ユーザプログラムの持つバッファは固定長である必要があるので、固定長のバッファを用意し繰り返し使用している。従来技術を用いたデータ転送では、ユーザプログラムの持つバッファの大きさをどのように設定しても、データはOSが持つシステムバッファのサイズごとに分割されて転送されるので、分割処理あるいは統合処理のオーバーヘッドが存在する。一方本発明を用いたデータ転送ではユーザプログラムの持つバッファ単位でデータ転送が行われ、ユーザプログラムによる分割処理あるいは統合処理は依然として存在するものの、ユーザプログラムの扱うデータの規模を考慮した適当なサイズのユーザバッファを利用することで、分割処理あるいは統合処理はユーザプログラムによる最小限の処理のみとなる。さらに、本実施例におけるファイル転送システムのような応用の場合では、直接ファイル入出力を行うため、データは分割処理あるいは統合処理は全く不要となる。各送信側ユーザプログラム103は、送信受付プログラム106を呼び出す際に、送信データが格納されているバッファ104の先頭アドレスと、バイト単位で示したファイルの大きさを表す送信サイズと、通信相手を選定するための通信路番号を引数として渡す。通信路番号は、特定の送信側ユーザプログラム103と特定の受信側ユーザプログラム203との組み合わせを識別するために用いるもので、システム内で一意な数値であり、TCP/IPにおけるコネクション識別子に相当する。通信路はTCP/IPと同様に、通信を行う二つのユーザプログラムのそれぞれが明示的に通信路(コネクション)の終端(ソケット)を開設し、一方の終端に公知の(お互いが既知な)名前をつけ、もう一方の終端が名前付終端への接続を要求し、名前付終端が接続を許可することで成立する。通信路識別子(コネクション識別子)は二つの終端の組を一意に識別できるのであり、ユーザプログラムは複数個の通信路あるいは終端を作成し、あるいは使用できるが、本実施の形態に

においては二つのユーザプログラムの組が一つの通信路を扱っている。以下では、三つの送信側のユーザプログラム103の持つ通信路番号はそれぞれ1001、1002、1003であると仮定する。また、三つの送信側のユーザプログラム103のそれぞれの通信相手となる受信側の三つのユーザプログラム203の持つ通信路番号も同一の通信路番号を持つ。送信サイズはゼロより大きくなければならない。

【0026】(A2)送信受付プログラム106による送信要求の受付

図4のフローチャートに示すように、送信受付プログラム106は、呼び出されると、まず新しい送信要求アイテム510(図5)を作成し(ステップ411)、その中の項目を埋める(ステップ412)。図5に示すように、送信要求アイテム510は、主記憶装置102の送信データを保持したバッファの先頭のアドレスを示す送信開始アドレス511と、どれだけのデータを送信するかを示す送信サイズ512と、送信先を選定する通信路番号513と、だれが送信したかを示す要求元プログラム識別子514と、送信要求アイテム510を連結するための次ポインタ515からなる。新しい送信要求アイテム510の送信開始アドレス511と送信サイズ512と通信路番号513にはユーザプログラム103からの呼び出し時に引数で与えられた情報を格納し、要求元プログラム識別子514には呼び出し元のユーザプログラム103のプログラム識別子を格納する。

【0027】その後、送信受付プログラム106は、作成した送信要求アイテム510を送信要求リスト108に追加する(ステップ413)。すなわち、作成した送信要求アイテム510の次ポインタ515には、送信要求リスト108の中で最も新しい送信要求アイテム510の先頭アドレスを格納する。なお、新しい送信要求アイテム510を送信要求リスト108に追加する際には、送信要求リスト108内のそれぞれの送信要求アイテム510を調べて、その中に同一の通信路番号513を持ち、かつ送信サイズ512がゼロである送信要求アイテム510が存在するか否かを調べる。そのような送信要求アイテム510は後述する送信失敗の報告に用いられ、この時点ではすでに用済みであるので送信要求リスト108から削除する。

【0028】その後、送信受付プログラム106は、現在実行中の処理を凍結して再び処理が解凍されるまで休止することで送信完了報告を待つ(ステップ414)。送信受付プログラム106はユーザプログラム103により呼び出されており実行中はユーザプログラム103のプログラム識別子を持つ。休止するのは「送信受付プログラム106を呼び出し中のユーザプログラム103」であり、あるユーザプログラム103が休止している最中に、別のユーザプログラム103が送信受付プログラム106を呼び出した場合には、送信受付プログラ

ム106は後者のユーザプログラム103のプログラム識別子を持って動作する。本実施の形態では、再開の指示は、後に説明する送信処理プログラム107が行う。

【0029】(A3)送信処理プログラム107(その1)REQコマンドの送信

図6に示すように、送信処理プログラム107はまず、送信要求リスト108を定期的に調べ(ステップ431、432)、新しい送信要求アイテム510があればそれを取り出し送信要求リスト108から削除する(ステップ433)。その際に送信サイズ512がゼロである送信要求アイテム510は、後述の送信失敗報告のための送信要求アイテムであるので無視する。取り出した送信要求アイテム510の各々に対してREQコマンドを作成し、受信側の計算機200に送る(ステップ434)。

【0030】具体的には、送信処理プログラム107は、ステップ434では、複数の送信要求アイテムからREQコマンドブロック109を作成する。REQコマンドブロック109は、図7(a)に示すように、それぞれ一つの送信要求アイテムに対して一つのREQコマンドを含む。各REQコマンドは、コマンドコード531と転送サイズ532と通信路番号533を有する。本実施の形態での具体的な例として、送信側の三つのユーザプログラム103から新たに発行された三つのREQコマンドが、送信処理プログラム107でのステップ431、432の実行時に検出されたと仮定する。図7(a)に示すように、これらの三つのREQコマンドは、それぞれ通信路番号533として、1001、1002、1003を有し、転送サイズ532として300、300、500を有すると仮定する。

【0031】送信処理プログラム107は、REQコマンドブロック109を送信するために、さらにDMA制御リスト111に送信命令を書いて、通信装置112を起動する。DMA制御リスト111は、図7(b)に示すように、それぞれ一つの送受信命令に対応する複数のエントリからなり、各エントリは、送信命令か受信命令のいずれかであるかを示す送受信命令501と、送受信命令501が送信命令の場合に、送信すべきデータの主記憶装置102上の先頭アドレスを表し、あるいは送受信命令501が受信命令の場合に、受信すべきデータを格納すべき主記憶装置102上の先頭アドレスを表す送受信開始アドレス502と、送信データあるいは受信データのサイズを表す送受信サイズ503を含む。今の場合、REQコマンドブロック109を送信するには、このDMA制御リスト111の一つのエントリとして、送受信命令501に送信命令を、送受信開始アドレス502にREQコマンドブロック109の先頭アドレスを、送受信サイズ503にはREQコマンドブロック109の大きさ(固定長)をそれぞれ格納する。

【0032】(A4)通信装置112(その1)

(A41)REQコマンドブロック109の送信

通信装置112は、起動されると、命令制御回路115が、DMA制御リスト111の先頭のエントリを主記憶装置102から読み出し、今の場合には、このエントリには送信命令が含まれているので、送信動作を開始する。すなわち、この先頭のエントリに含まれた送受信開始アドレス502と送受信サイズ503に従い、REQコマンドブロック109に含まれた複数のREQコマンドを連続して主記憶装置102から読み出すようDMA回路113に指示し、通信回路114に、この読み出されたREQコマンドブロック109をチャンネル300を介して直ちに計算機200の通信装置212に送るよう指示する。命令制御回路115は、DMA制御リスト111の後続のエントリに有効な命令があればそれに応答するが、今の場合にはDMA制御リスト111そのような後続の有効な命令はない。

【0033】(A42)送信完了通知

命令制御回路115は、この送信を終えるとハードウェア割り込みを発生し、送信処理プログラム107に送信完了を通知する。送信処理プログラム107はこの割り込みを検知することで送信完了を確認する。

【0034】(B)受信側計算機200(その1)

(B1)受信側ユーザプログラム203(その1)受信要求の発行

本実施の形態では、送信側ユーザプログラム103がOS105に対して送信要求を発行するとともに、受信側のユーザプログラム203がOS205に対して受信要求を発行するように構成され、後に説明するように、送信要求を発行したユーザプログラム103からの送信データは、いずれかの送信先のユーザプログラム203がそのデータを受信可能な状態になった後に、具体的には、送信先のユーザプログラム203がそのデータに対して受信要求を発行した後にのみその送信先のユーザプログラムに転送される。

【0035】図8のフローチャートに示すように、受信側ユーザプログラム203の処理も送信側ユーザプログラム103の処理と同様の2重のループ構造をなしており、処理の最初と最後に転送されたファイルを保持するための受信用のパバッファ204を確保し、あるいは解放する(ステップ451、458)。外側のループは複数のファイルを処理するためのループであり、内側のループは各々のファイルを複数回の通信で転送するためのループである。

【0036】各受信側ユーザプログラム203は、受信受付プログラム206を呼び出し、受信要求を通知し(ステップ452)、パバッファ204に格納されたデータをファイルの一部として格納する(ステップ455)。

【0037】受信受付プログラム206を呼び出しから戻ったときは、呼び出しからの戻り値を調べてエラーで

10

20

30

40

50

あることがわかるとエラー処理を行い終了する（ステップ454）。ファイルへの出力が済むと、現在処理中のファイルが全て転送済みであるか調べて未処理の領域があればそれらを処理するためにステップ452から繰り返す（ステップ456）。各ユーザプログラムは送信すべきファイルがまだある限り以上のステップを繰り返す（ステップ457）。

【0038】（B2）受信受付プログラム206による受信要求の受付

図9のフローチャートに示すように、受信受付プログラム206は、呼び出されると、まず新しい受信要求アイテム520を作成し（ステップ421）、その中の項目を埋める（ステップ422）。受信要求アイテム520は、図10に示すように、受信したデータを主記憶装置202のどのアドレス位置から格納するかを示す受信開始アドレス521と、どれだけかのデータが受信可能かを示す最大受信サイズ522と、送信元のユーザプログラムとその受信要求を発行したユーザプログラムの組に対して定められた通信路番号523と、受信要求元のユーザプログラムの識別子524と、他の受信要求アイテム520と連結するための次ポインタ525からなる。作成した新しい受信要求アイテム520の受信開始アドレス521と最大受信サイズ522と通信路番号523には受信要求元のユーザプログラムからの呼び出し時に引数で与えられた情報を格納し、要求元プログラム識別子524には呼び出し元である受信側ユーザプログラム203のプログラム識別子を格納する。なお、最大受信サイズ522はゼロよりも大きくなければならない。

【0039】受信受付プログラム206は、作成した受信要求アイテム520を受信要求リスト208に追加する（ステップ423）。すなわち、作成した受信要求アイテム520の次ポインタ525には、受信要求リスト208の中で最も新しい受信要求アイテム520のアドレスを格納する。なお、新しい受信要求アイテム520を受信要求リスト208に追加する際には、受信要求リスト208内のそれぞれの受信要求アイテム520を調べて、その中に同一の通信路番号523を持ち、かつ最大受信サイズ522がゼロである受信要求アイテム520が存在するか否かを調べる。そのような受信要求アイテム520は後述する受信失敗の報告に用いられ、この時点ではすでに用済みであるので受信要求リスト208から削除する。

【0040】その後、受信受付プログラム206は、送信受付プログラム106と同様に現在実行中の処理を凍結して再び処理が解凍されるまで休止することで受信完了報告を待つ（ステップ424）。本実施の形態では、再開の指示は、次に説明する受信処理プログラム207が行う。

【0041】（B3）受信処理プログラム207（その1）REQコマンドブロックの受信命令

受信側の計算機200では、送信側の計算機100から送信されるREQコマンドブロックをいつでも受信できるように準備をしておく必要がある。受信処理プログラム207は、このコマンドブロックの受信を通信装置212に命令する。

【0042】すなわち、図11のフローチャートに示すように、受信処理プログラム207は、REQコマンドブロックを受信し、主記憶装置202にREQコマンドブロック209として格納することを指示する受信命令をDMA制御リスト211に書いて、通信装置112を起動し、受信完了を待つ（ステップ441）。

【0043】図12（b）に示すように、DMA制御リスト211は、DMA制御リスト111と同じ構造を有する。受信処理プログラム207は、REQコマンドブロック109を受信してREQコマンドブロック209として主記憶装置202として格納するために、DMA制御リスト211の一つのエントリに、送受信命令504として受信命令を、送受信開始アドレス505としてREQコマンドブロック209の先頭アドレスを、送受信サイズ506としてREQコマンドブロック209の大きさ（固定長）をそれぞれ格納する。

【0044】（B4）通信装置212（その1）

（B41）REQコマンドブロック109の受信

通信装置212が起動されると、命令制御回路215がDMA制御命令を解釈し、実行する。まず、命令制御回路215が、DMA制御リスト211の最初のエントリを主記憶装置202から読み出し、このエントリ中の命令、今の場合には受信命令に従って通信回路214およびDMA回路213を制御する。すなわち、通信回路214にコマンドブロックの大きさのデータの受信を命じ、通信回路214がREQコマンドブロック109の内容をチャンネル300を介して計算機100から受信すると、DMA回路213に、受信したREQコマンドブロック109の内容を主記憶装置202上のREQコマンドブロック209に直ちに書き込ませる。すなわち、上記読み出したエントリに含まれた送受信開始アドレス505と送受信サイズ506に従い、受信したREQコマンドブロック109を主記憶装置202にREQコマンドブロック209として書き込む。図7（b）には、三つの送信要求REQを含むREQコマンドブロック209の例を示している。DMA回路213は、DMA制御リスト211の後続のエントリに有効な命令があればそれに応答するが、今の場合にはDMA制御リスト211そのような後続の有効な命令はない。

【0045】（B42）受信完了通知

命令制御回路215は、この書き込みが終了するとハードウェア割り込みを発生し、受信処理プログラム107に受信完了を通知する。受信処理プログラム107はこの割り込みを検知することにより受信完了を確認し、次のステップ441に進む。

【0046】(B5)受信処理プログラム207(その2)

(B51)ACK/NACKコマンドの送信命令
REQコマンドブロック209の受信が完了すると、受信処理プログラム207は、受信したREQコマンドブロック209内のそれぞれのREQコマンドの対して、受信が可能であるか否かを調べ、ACKコマンドまたはNACKコマンドを作成し、それらのコマンドを含むACKコマンドブロックを送信側の計算機100に送る(ステップ442)。

【0047】すなわち、それぞれのREQコマンドが受信可能か否かを調べるために、受信要求リスト208から、REQコマンドブロック209内のそれぞれの各REQコマンドに対する通信路番号533と同じ通信路番号523を持つ受信要求アイテム520を探す。該当する受信要求アイテム520が存在しなければ、そのREQコマンドは受信不可能である。すなわち、REQコマンドブロック209の受信が完了した時点で、その通信路番号533を指定する受信要求が受信側のユーザプログラム203のいずれかによりまだ発行されていない場合には、このREQコマンドは受信不可能となる。上記該当する受信要求アイテム520が存在した場合には、さらにそのREQコマンドに対する転送サイズ532がその該当する受信要求アイテム520の最大受信サイズ522より大きくないかを判別する。その転送サイズ532がその最大受信サイズ522を超えていればそのREQコマンドは受信不可能であるが、そうでなければそのREQコマンドは受信可能である。なお、このステップ442においては、最大受信サイズ522がゼロである受信要求アイテム520は、後述の受信失敗報告のための受信要求アイテムであるので無視する。受信不可能なREQコマンドには2種類あり、単に対応する受信要求が未発生の場合はREQコマンドの通知が早すぎただけなのでそのREQコマンドは送信側計算機100において再び処理を試みられるが、サイズ超過の場合はプログラムミスなどの不測の事態であり、送信要求および受信要求は失敗としてそれぞれのユーザプログラムに通知される(この場合には受信要求はすでに発生している)。

【0048】本実施の形態では、具体的な例示に当たり、送信側の三つのユーザプログラム103から図7(a)に例示する三つのREQコマンドが発行された後、REQコマンドブロック209の受信が完了した時点までに、これらの三つのREQコマンドの内の二つに対応する受信要求が受信側の二つのユーザプログラム203から発行され、これら二つのREQコマンドのそれぞれが指定する転送サイズ532は、それらの対応する受信要求が指定する最大受信サイズを超えないと仮定する。さらに、上記三つのREQコマンドの内の第3のREQコマンドに対応する受信要求は受信側の他の一つの

ユーザプログラムからは遅れて発行されるが、その第3のREQコマンドが指定する転送サイズ532もその対応する第3の受信要求が指定する最大受信サイズを超えないと仮定する。この場合には、REQコマンドブロック209の受信が完了した時点では、最初の二つのREQコマンドが受信可能となり、この最後のREQコマンドは受信不可能となる。

【0049】こうして受信可能と判断されたREQコマンドおよび受信不可能と判断されたREQコマンドに対して、それぞれACKコマンドおよびNACKコマンドを生成し、ACKコマンドブロック210として計算機100に送る。図12(a)に示すように、ACKコマンドブロック210には、REQコマンドブロック209内の受信可能と判断されたREQコマンドに対しては、そのREQコマンドブロック209内のコマンド部531にあるREQを「ACK」に変更しただけのACKコマンドが含まれる。さらに、REQコマンドブロック209内の受信不可能と判断されたREQコマンドに対しては、そのREQコマンドブロック209内のコマンド部531にあるREQを「NACK」に変更するとともに、受信不可能の原因が転送サイズの超過である場合は、その転送サイズ532を受信可能なサイズに変更する。送信側の計算機ではこの転送サイズ532の変更の有無を調べることによって、受信不能の理由が受信の未準備によるものかサイズ超過によるもの(致命的エラー)かを判断できる。

【0050】図12(a)の例では、ACKコマンドブロック210は、二つのACKコマンドと一つのNACKコマンドからなる。すなわち、通信路番号が1001および1002であるREQコマンドは、対応する受信要求がすでに発生しており、要求される転送サイズも受信可能な量を超えていないので、受信可能でありACKコマンドが返される。一方、通信路番号が1003であるREQコマンドは、対応する受信要求が未発生なために受信不可でありNACKコマンドが返される。この受信不可の原因は転送サイズ超過ではなく受信要求未発生であるので、NACKコマンドの転送サイズ532は変更されない。

【0051】受信処理プログラム207は、生成したACKコマンドブロック210を計算機100に送信する命令を発行する。その命令は、計算機100におけるREQコマンドブロック109の送信と同様である。すなわち、受信処理プログラム207は、DMA制御リスト211内に、送受信命令504として送信命令を、送受信開始アドレス505としてACKコマンドブロック210の先頭アドレスを、送受信サイズ506としてACKコマンドブロック210の大きさ(固定長)をそれぞれ有する送信命令を格納し、通信装置212を起動する。その後、送信完了を確認するために、通信装置212からのハードウェア割り込みを待つ。

【0052】(B52) 受信失敗報告

以上の仮定的な例では発生していないが、あるREQコマンドに対して通信路番号が一致する受信要求がすでに受信側のいずれかのユーザプログラムからすでに発行されているながら、そのREQコマンドが指定する転送サイズがその受信要求が指定する最大受信サイズを超過した場合、受信処理プログラム207は、その受信要求を発行した受信側ユーザプログラム203に受信失敗を報告する。すなわち、受信処理プログラム207は、受信に失敗した受信要求アイテム520の最大受信サイズ522をゼロに変更し、変更後の受信要求アイテム520を受信要求リスト208に戻し、さらにその受信要求アイテム520の要求元プログラム識別子524で特定される受信完了待ちの受信側ユーザプログラム203を再起動する(ステップ444(図11))。

【0053】受信受付プログラム206は、(ユーザプログラム203の呼び出し延長処理として)処理を再開されると、受信要求リスト208内に自分のプログラム識別子(この場合は呼び出し元のユーザプログラム203のプログラム識別子)を持った受信要求アイテム520が存在するかどうかを調べて、受信側ユーザプログラム203に戻る。返り値には、受信要求が戻されていれば失敗を表す返り値を、なければ成功を表す返り値を用いる(ステップ425(図9))。

【0054】(B6) ユーザプログラム203(その2)

受信受付プログラム206が終了すると呼び出し元のユーザプログラム203に処理が戻る。ユーザプログラム203は呼び出しの結果の戻り値を調べて(ステップ453(図8))、エラーであればエラー処理を行い終了する(ステップ454(図8))。

【0055】以上の仮定的な例では呼び出しは成功裡に終わりエラー処理は起こっていない。

【0056】(C) 送信側計算機(その2)

(C1) 送信処理プログラム107(その2) ACKコマンドブロックの受信命令

送信側の計算機100ではACK/NACKコマンドブロック210を受け取る準備を行う。すなわち、図8のステップ435に示すように、送信処理プログラム107は、DMA制御リスト111に、ACKコマンドブロックの受信命令を書いて、通信装置112を起動し、受信完了を待つ。この命令は、ACKコマンドブロックを受信し、主記憶装置102にACKコマンドブロック110として書き込むことを要求する。具体的には、送信処理プログラム107は、DMA制御リスト111に、送受信命令501に受信命令を、送受信開始アドレス502にACKコマンドブロック110の先頭アドレスを、送受信サイズ503にはACKコマンドブロック110の大きさ(固定長)をそれぞれ格納する。受信完了の確認は、通信装置112からの割り込みを待つことで

行う。

【0057】(C2) 通信装置112

通信装置112は、この受信命令に回答して、ACKコマンドブロック210を受信し、主記憶装置102にACKコマンドブロック110として書き込む。このときのこの回路の動作は、通信装置112がREQコマンドブロックを受信した場合と同じである。

【0058】(C3) 送信処理プログラム107(その3) データ送信命令

10 送信側の計算機100は、ACKコマンドブロックを受け取るとデータの送信を開始する。すなわち、送信処理プログラム107は、図8のステップ436において、ACKコマンドブロック110に含まれた複数のACKコマンドをそれぞれ取り出し、それぞれに対応する送信データの送信を通信装置112に命令し、これを起動し、そのデータの送信の完了を待つ。各ACKコマンドに対する送信データは、そのACKコマンドと同じ通信路番号を有する送信要求アイテム510を送信要求リスト108から検索し、その送信要求アイテム510内の送信開始アドレス511、送信サイズ512でもって指定する。すなわち、送信処理プログラム107は、DMA制御リスト111内の送受信命令501に送信命令を、送受信開始アドレス502に検索された送信開始アドレス511を、送受信サイズ503に検索された送信サイズ512をそれぞれ格納することでもって、そのACKコマンドに対するデータの送信を通信装置112に命令する。

30 【0059】送信処理プログラム107は、ACKコマンドブロック110に複数のACKコマンドがある場合には、それらに対応する複数の送信命令をDMA制御リスト111に格納する。それらの送信命令をDMA制御リスト111に格納する順序は受信されたACKコマンドブロック110内に格納されていた対応するACKコマンドの順序と同じにする。

【0060】送信処理プログラム107は、受信されたACKコマンドブロック110内にNACKコマンドがある場合には、それぞれのNACKコマンドに対して2通りの処理のいずれか一方を行う。すなわち、各NACKコマンド内の転送サイズ532の値が、対応するREQコマンドが要求した送信サイズ、すなわち、そのREQコマンドに対応する送信要求アイテム510内の送信サイズ512に等しければ、受信不可の原因は受信側計算機200で受信要求が未だ発生していないことにある。従って、送信処理プログラム107は、REQコマンドを再送するために、送信要求アイテム510を送信要求リスト108に戻す。

50 【0061】一方、そのNACKコマンド内の転送サイズ532が対応するREQコマンドが要求した送信サイズより縮小されていれば、受信不可の原因は、対応するREQコマンドが要求した送信サイズが、受信側の計算

機200で発生した受信要求が指定した最大受信サイズを超過しているためである。この場合は、送信処理プログラム107は送信側ユーザプログラム103に送信失敗の報告を行う。すなわち、送信処理プログラム107は、失敗した送信要求アイテム510の要求元プログラム識別子514で特定される送信完了待ちの送信側ユーザプログラム103の処理を再開する。その際に、失敗した送信要求アイテム510の送信サイズ512をゼロに変更し、送信要求リスト108に戻す。

【0062】(C4) 通信装置112 (その3) データ 10 送信

通信装置112は起動されると、送信処理プログラム107により命令されたデータの送信をDMA制御リスト111に基づいて行う。すなわち、DMA回路113は、DMA制御リスト111に記載された各送信命令が指定する送信データをユーザプログラム103内のバッファ104からDMA回路113が読み出し、通信回路114が送信し、通信回路114は、送信が完了すると割り込みにより送信処理プログラム107にそれを通知する。命令制御回路115はDMA制御リスト111に 20 複数のデータの送信命令が記載されているときには、それらの命令の記載順に従ってそれらの送信命令が指定する複数の送信データを順次読み出し、通信回路114がそれらのデータを順次送信する。このように、送信側では、送信データはユーザプログラム103のバッファからOS105にコピーされることなく送信される。また、ユーザプログラム103内の送信データは分割されることなく、連続して送信される。今の例の場合、二つのデータを送信し、二つの送信を終えるとハードウェア割り込みを発生させる。

【0063】(C5) 送信完了報告

送信処理プログラム107は、データ送信の完了を通信装置112から通知されると、送信側ユーザプログラム103に送信の完了報告を行う(ステップ437(図6))。図12(a)に示されたACKコマンドブロック209を送信した後では、二つのACKコマンドに対する二つのユーザプログラム203に送信完了の報告を行う。この際、送信の失敗報告と異なり、対応する送信要求アイテム510を送信要求リスト108に戻さない。

【0064】(C6) ユーザプログラム103 (その2) 送信失敗に対する処理

送信側のユーザプログラム103は送信受付プログラム106の呼び出しから返ってくると、その戻り値を調べ(ステップ404(図3))、エラーであればエラー処理を行う(ステップ405(図3))。

【0065】(D) 受信側計算機200 (その2)

(D1) 受信処理プログラム207 (その3) データ受信命令

受信処理プログラム207は、ACKコマンドブロック 50

の送信命令を発行後、そのACKコマンドブロックによりACKコマンドを一つでも返信していれば(ステップ443(図9))、そのACKコマンドに対応するデータの送信命令の送信を通信装置212に命令し、通信装置212を起動し、その送信の完了を待つ(ステップ444(図9))。ACKコマンドに対応して受信すべきデータは、そのACKコマンドに含まれる通信路番号を有する受信要求アイテム520を受信要求リスト208から検索し、その受信要求アイテム520内の受信開始アドレス521で指定されるデータである。従って、送信処理プログラム107は、DMA制御リスト211内の送受信命令コード504に受信命令コードを、送受信開始アドレス505に検索された受信開始アドレス521を、送受信サイズ506に返したACKコマンドで指定した転送サイズ532をそれぞれ格納することでもって、そのACKコマンドに対するデータの受信を通信装置212に命令する。そのACKコマンドブロック210に複数のACKコマンドが格納されていた場合には、それらのACKコマンドの順序に従って、それぞれに対応するデータ受信命令をDMA制御リスト211に格納する。

【0066】なお、送信処理プログラム107は、ステップ443の判定の結果、送信したACKコマンドブロック210によりACKコマンドを一つも返さなかったと判定されたときには、データ受信命令の発行は行わずにステップ441に戻り、もう一度REQコマンドブロックを受けるための命令を発行する。

【0067】(D2) 通信装置211 (その3)

(D21) データの受信

30 通信装置212は起動されると、受信処理プログラム207により命令されたデータの受信をDMA制御リスト211に基づいて行う。すなわち、命令制御回路215は、DMA制御リスト211の先頭のエントリを主記憶装置202から読み出し、通信回路214がデータを送信側計算機100から受信すると、読み出したエントリに含まれた受信命令が指定する、いずれかのユーザプログラム103内のバッファ104にその受信されたデータを書き込む。DMA制御リスト211に複数のデータの受信命令が記載されているときには、命令制御回路2 40 15は、それらの命令の記載順に従って、順次受信された複数のデータをそれらの受信命令が指定する、複数のユーザプログラム203内のバッファ204に順次書き込む。図12(a)に示されたACKコマンドブロックを送信した後では、二つのACKコマンドに対する二つのデータが受信される。このように、受信側でも、通信装置214により受信されたデータは、従来技術のようにOS205内のバッファにコピーされることなく、直接いずれかのユーザプログラム203のバッファ204に書き込まれる。

【0068】(D22) 受信完了通知

命令制御回路215は、いずれかの受信データの書き込みを完了すると、ハードウェア割り込みを発生し、受信処理プログラム207にデータ受信の完了を通知する。受信処理プログラム107はこの割り込みを検知することにより受信完了を確認し、次のステップ445に進む。なお、DMA制御リスト211に複数のデータの受信命令が記載されているときには、それらの命令が指定する複数のデータの各々をいずれかのバッファ203に書き込むごとに、上記受信完了割り込みを発生する。

【0069】(D3) 受信処理プログラム207 (その4)

(D31) 受信完了報告

通信装置211によりデータ受信完了の通知がなされると、受信処理プログラム207は、受信したデータに対応するACKコマンドに対応する受信要求元のユーザプログラム203に受信の完了報告を行う(ステップ445(図9))。図12(a)に示されたACKコマンドブロックを送信した後では、二つのACKコマンドに対する二つのユーザプログラム204に受信完了の報告を行う。この際、対応する受信要求アイテム510を受信要求リスト108に戻さない。

【0070】(D32) 受信側ユーザプログラム203
受信側のユーザプログラム203は受信受付プログラム206の呼び出しから戻ると戻り値を調べる。失敗の場合はACKコマンドブロック210を作成する時点で戻って来、この例のこの時点では必ず成功裡に戻ってくる。戻ってくるとバッファ204に受信したデータをファイルの一部に出力し(ステップ455(図8))、出力が済むと、現在処理中のファイルが全て転送済みであるか調べて未処理の領域があればそれらを処理するためにステップ452から繰り返す(ステップ456)。各ユーザプログラムは送信すべきファイルがまだある限り以上のステップを繰り返す(ステップ457)。

【0071】(E) NACKコマンドに対するデータの転送

今考えている仮定的な例では、通信路番号が1003である送信要求アイテム510に対しては、対応する受信要求が受信側の計算機200で出力されなかったためにNACKコマンドがそこから返送された。このため、この送信要求アイテム510に対する送信データは送信されなかったが、このデータは以下のようにしてその後送信される。

【0072】図8に示すように、送信処理プログラム107は、受信したACKコマンドに対して前述した送信命令を発行した後、ステップ432へ処理を移して新たな送信要求を探す。前述したデータを送信しなかった送信要求アイテム510は送信処理プログラム107により、送信要求リスト108に戻されている。従って、送信処理プログラム107が新たな送信要求を探す時点で、この送信要求アイテム510が探索され、先に記

載した方法により対応するREQコマンドを含むREQコマンドブロックが計算機200に送られる。その時点で、受信側の計算機200で、通信路番号が1003である受信要求がすでに発行されているときには、受信側の計算機200からこのREQコマンドに対するACKコマンドを含むACKコマンドブロックが先に記載した方法により返送され、その後、このREQコマンドに対する送信データが計算機100から送信され、計算機200により受信される。こうして、先に送信失敗した送信データも送信することができる。

【0073】以上のように、本実施の形態によれば、ユーザプログラムのバッファとOSのバッファの間のデータコピー処理が送信側計算機と受信側計算機のいずれにおいても必要でないため、計算機間データ転送を高速に行える。

【0074】従来のTCP/IPによるデータ転送では、受信側ユーザプログラムの準備が整ってなくてもデータは送信される。その場合には、受信側の計算機において一旦受信データをOSバッファに蓄えてから受信側ユーザプログラムのバッファにコピーする必要がある、このデータコピー処理のオーバーヘッドが高速なデータ転送の実現を妨げていた。また、受信準備が整っていない受信側ユーザプログラムが多すぎるとOSバッファが溢れてデータを再送しなければならず、この再送に備えて送信側の計算機でもOSバッファに一旦データを蓄えておく必要があった。このため送信側の計算機でもデータコピーが発生し、そのオーバーヘッドが高速なデータ転送の実現を妨げていた。

【0075】しかるに、本実施の形態によるデータ転送ではデータ転送に先立つコマンドのやりとりで受信準備の完了を確認し、受信準備が整っていない場合にはデータそのものでなくコマンドのみを再送する方法を採ることにより、従来のデータコピーを不要にしている。

【0076】また、本実施の形態では、OSの固定長の通信バッファの大きさを超える大きなデータを転送する場合にもデータを分割して転送する必要があるため、通信処理のオーバーヘッドが小さく、データ転送を高速に行える。

【0077】従来のTCP/IPによるデータ転送では、事前に送受信するデータサイズを調停する手段を持たないため、転送サイズはOSの設定した固定値である必要がある。従って、これを超える大きなデータを転送する場合に、送信側のOSで転送サイズごとにデータを分割しておき、受信側のOSでは受けたデータを再構成する必要がある。このデータの分割、再構成処理のオーバーヘッドのために高速なデータ転送が実現できなかった。

【0078】しかるに、本実施の形態によるデータ転送では事前に送受信するデータサイズを調停することで任意の大きさのデータを転送できるので、データの分割、

統合処理は不要となり、高速なデータ転送が実現可能である。同様に通信装置が転送するデータの個数も事前に調停することで複数のデータ転送の通信を一括して処理できるので、通信装置の起動処理、後処理（割り込み処理）が削減され、さらに高速なデータ転送が実現可能である。

【0079】従来のHDL Cによるデータ転送では、本発明と同様にデータ転送に先立ってデータ転送の調停を行うので、データコピーのオーバーヘッドは存在しない。しかし、データ転送にOSが介在せず、転送を行うユーザプログラムは通信装置を独占してしまい、同時に複数のユーザプログラムがデータ転送を行えないという欠点があった。

【0080】しかるに、本実施の形態によるデータ転送では、調停作業をOSが一括して行うために、データコピーのオーバーヘッドがないという利点を保ったまま、同時に複数のデータ転送の調停を矛盾なく行うことができる。

【0081】＜変形例＞本発明は、以上の実施の形態に限定されるものではなく、いろいろの形態で実施可能である。例えば、

(1) 上に示した発明の実施の形態では、二つの計算機のみが接続されているシステムであったが、より多くの計算機が相互に接続されているシステムでもよい。この場合、異なる計算機対に対応した伝送路が各計算機対に対して設けられ、各計算機対がその対応する伝送路により接続されていることが望ましい。

【0082】(2) 計算機を接続する伝送路としては、実施の形態で示したチャンネルに限らず、例えば、複数の計算機を接続するインタコネクトネットワークでもよい。例えば、クロスバスイッチ、ハイパクロスバスイッチあるいは多段スイッチネットワークでもよい。

【0083】(3) これらの計算機間でのデータ転送は、パケットの形でデータを転送するものでもよい。例えば、上記ハイパクロスバネットワークで接続された複数の計算機間では通常はデータはパケットの形式で転送される。各パケットに転送先の計算機を指定するアドレスと転送すべきデータその他が含まれる。本発明は、このような形式でデータを計算機間で転送するシステムにも適用できる。

【0084】(4) 実施の形態では、ユーザプログラムとOS間の入出力インタフェースは同期型、すなわちユーザプログラムはOSに依頼した入出力処理が完了するまで待つタイプの入出力インタフェースであったが、非同期型の入出力インタフェースでもよい。実施の形態で

示したファイル転送システムを非同期入出力インタフェースを備えたファイル転送システムにするには、送受信の受付プログラムが送受信の完了を待たずに直ちにユーザプログラムに制御を戻し、ユーザプログラムは送受信の要求を発行した後、改めて送受信の完了を待つ要求を受付プログラムに発行し、受付プログラムはこれを受けて残りの処理を行えばよい。

【0085】

【発明の効果】以上の記載から明らかなように、本発明によれば、ユーザプログラムのバッファとOSのバッファの間のデータコピー処理がないため、計算機間のデータ転送を高速に行える。

【0086】さらに、大きなデータを転送する場合にもデータを分割して転送する必要がないため、通信処理のオーバーヘッドを小さくでき、データ転送を高速に行える。

【図面の簡単な説明】

【図1】本発明に係るデータ転送方法を実施する計算機システムの全体構成図。

【図2】図1の装置におけるデータ転送のタイムチャート。

【図3】図1の装置に使用する送信側ユーザプログラム(103)のフローチャート。

【図4】図1の装置に使用する送信受付プログラム(106)のフローチャート。

【図5】図1の装置に使用する送信要求リスト(108)のデータ構造を示す図。

【図6】図1の装置に使用する送信処理プログラム(107)のフローチャート。

【図7】(a)は、図1の装置に使用するREQコマンドブロック(109)のデータ構造を示す図。(b)は、図1の装置に使用するDMA制御リスト(111)のデータ構造を示す図。

【図8】図1の装置に使用する受信側ユーザプログラム(203)のフローチャート。

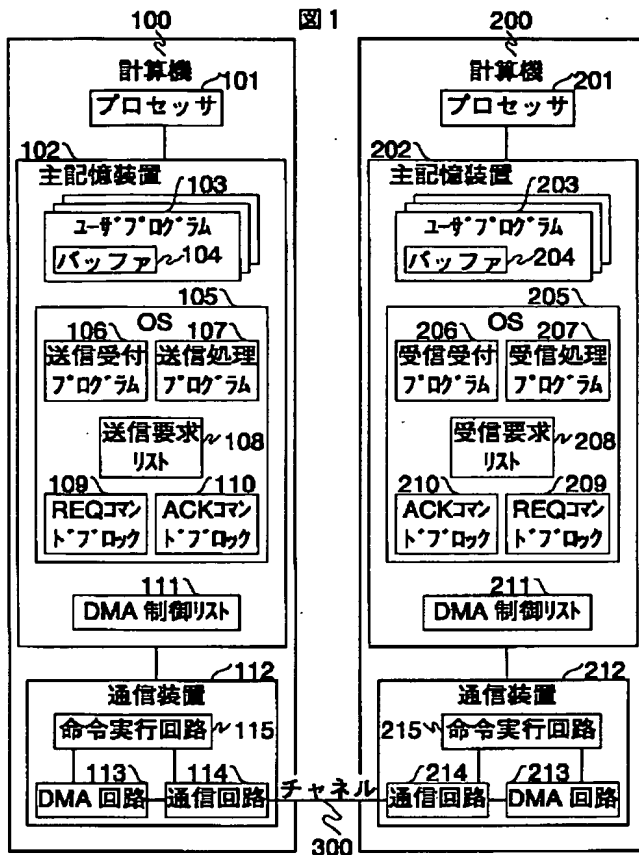
【図9】図1の装置に使用する受信受付プログラム(206)のフローチャート。

【図10】図1の装置に使用する受信要求リスト(208)のデータ構造を示す図。

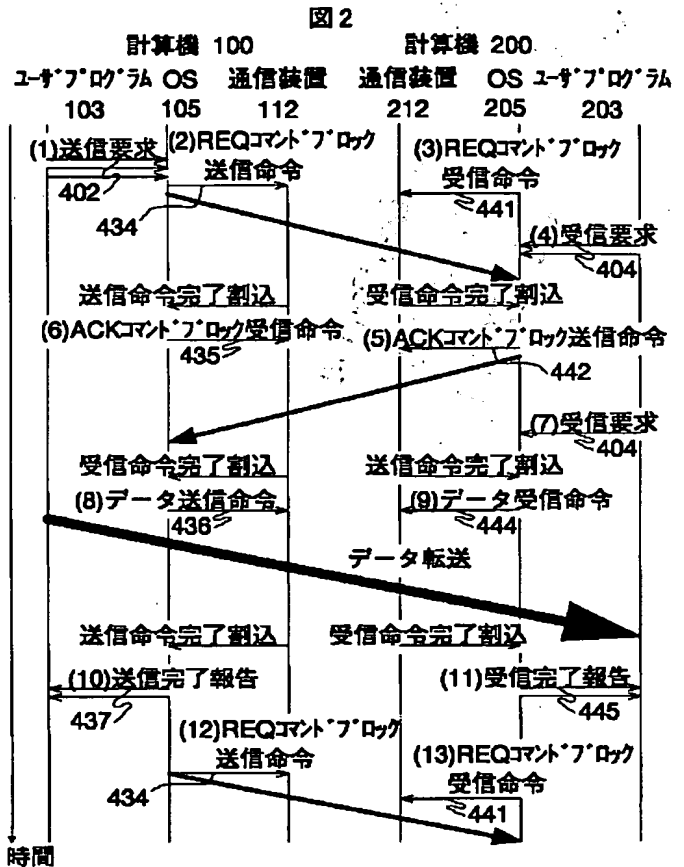
【図11】図1の装置に使用する受信処理プログラム(207)のフローチャート。

【図12】(a)は、図1の装置に使用するACKコマンドブロック(210)のデータ構造を示す図。(b)は、図1の装置に使用するDMA制御リスト(211)のデータ構造を示す図。

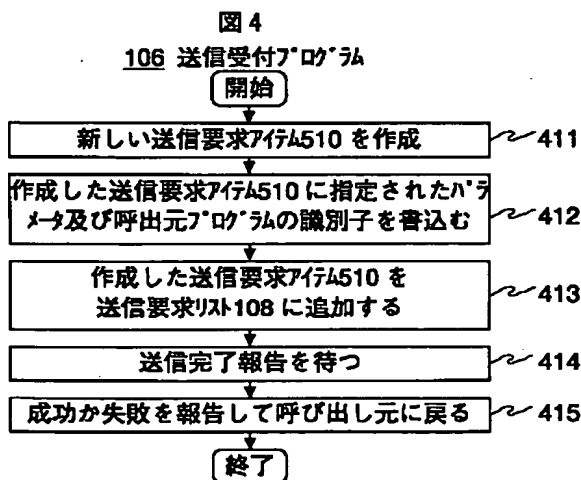
【図1】



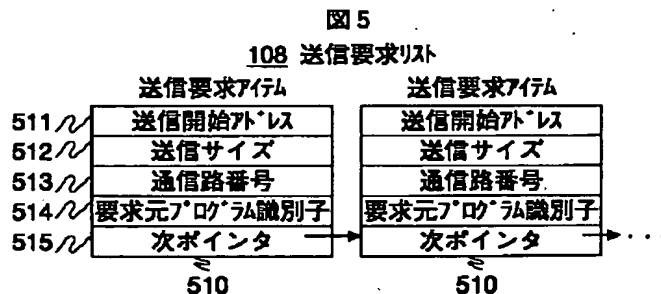
【図2】



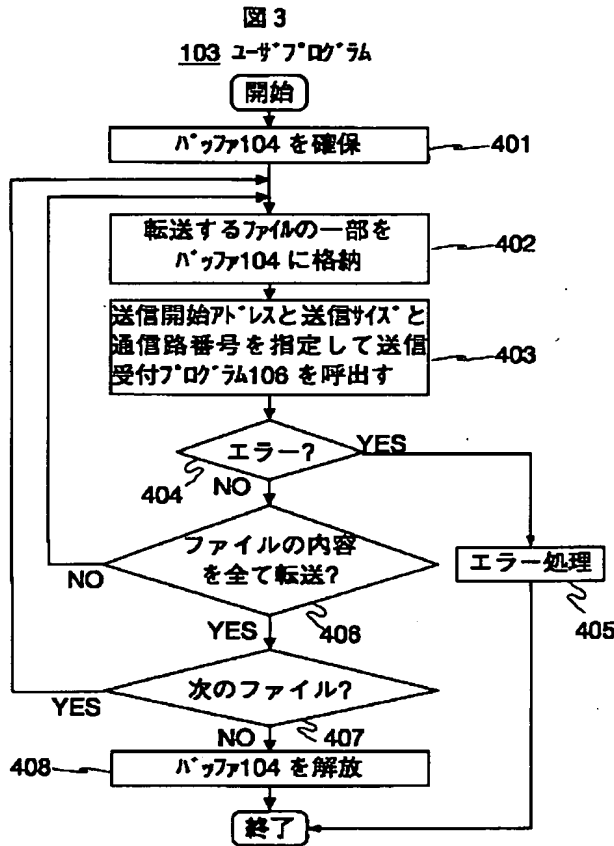
【図4】



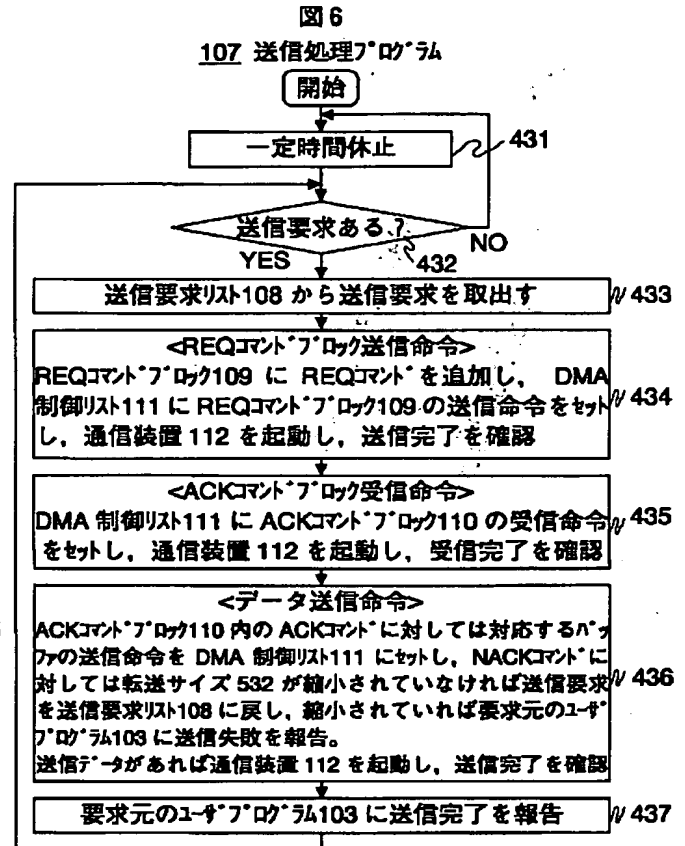
【図5】



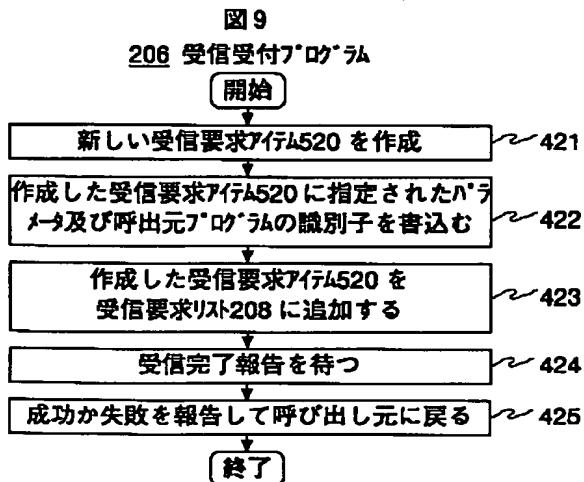
【図3】



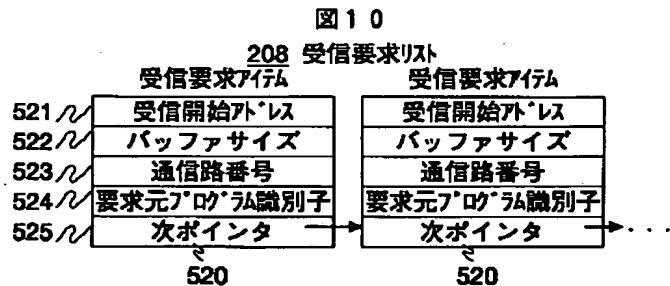
【図6】



【図9】



【図10】



【図7】

図7

(a)

109 REQコマンドプログラム

コマンド 転送サイズ 通信路番号
コード

REQ	300	1001
REQ	300	1002
REQ	500	1003

531

532

533

(b)

111 DMA 制御リスト

命令コード	送受信開始アドレス	送受信サイズ
SEND	A	a

501

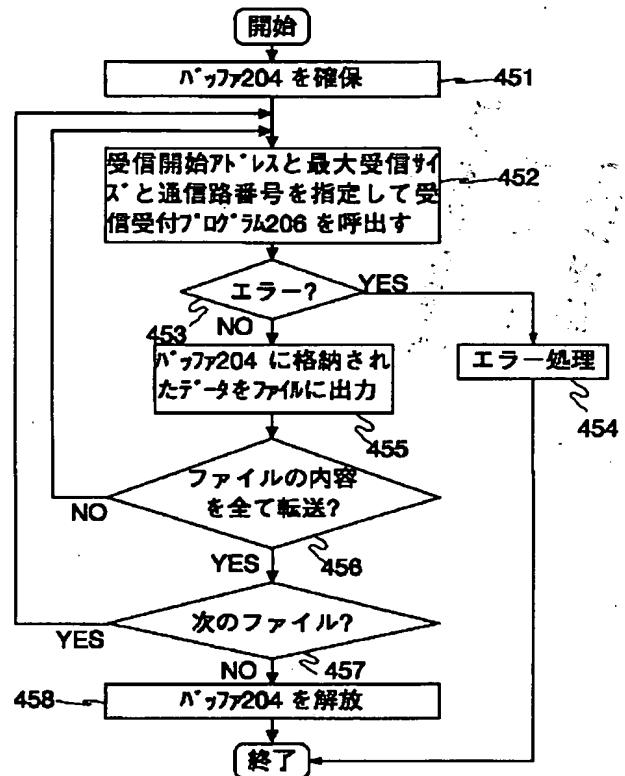
502

503

【図8】

図8

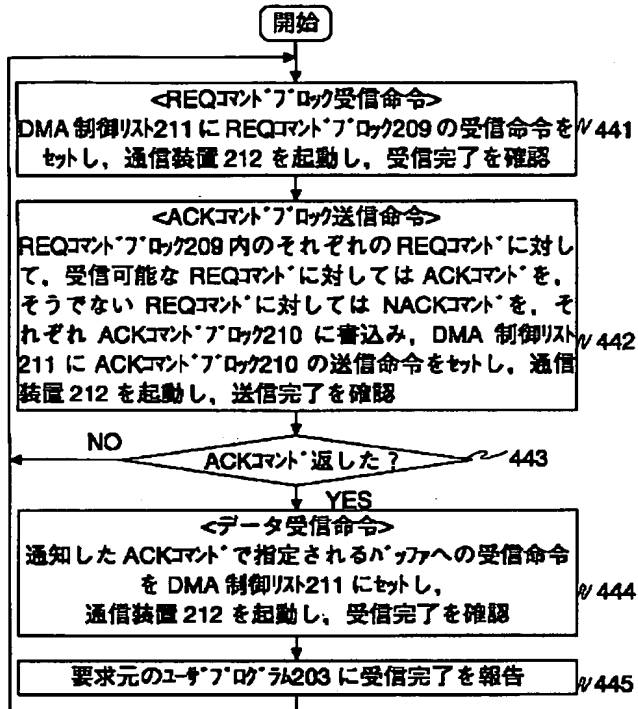
203 ユーザープログラム



【図11】

図11

207 受信処理プログラム



【図12】

図12

(a)

210 ACKコマンドブロック

コマンド
コード 転送サイズ 通信路番号

ACK	300	1001
ACK	300	1002
NACK	500	1003

531

532

533

(b)

211 DMA制御リスト

命令コード	送受信開始アドレス	送受信サイズ
SEND	B	b

504

505

506

フロントページの続き

(72)発明者 清水 正明
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内

(72)発明者 鍵政 豊彦
東京都国分寺市東恋ヶ窪一丁目280番地
株式会社日立製作所中央研究所内